



Tuomas Reijonen

QUONIAM-LASKUTUSMODUULI

QUONIAM-LASKUTUSMODUULI

Tuomas Reijonen

Opinnäytetyö

Syksy 2011

Tietotekniikan koulutusohjelma

Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, ohjelmistokehitys

Tekijä: Tuomas Reijonen
Opinnäytetyön nimi: Quoniam-laskutusmoduuli
Työn ohjaaja: Pertti Heikkilä
Työn valmistumislukukausi ja -vuosi: Syksy 2011
Sivumäärä: 37 + 2 liitesivua

Tässä opinnäytetyössä tehtiin laskutusmoduuli uuteen myyntiin tulevaan taloushallinto-ohjelmaan. Laskutusmoduuli on SQL-pohjainen työpöytäsovellus, jonka avulla voidaan tehdä tulostettava tai sähköinen lasku asiakkaalle. Laskutusmoduuli myös kirjaa ylös myyntitapahtumat, joita voidaan hyödyntää kirjanpidossa. Työssä myös laajennettiin olemassa olevaa admin-moduulia laskutusmoduulin käyttöön.

Työ toteutettiin C#-ohjelmointikielellä ja tiedonsäilytykseen käytettiin MSSQL-palvelinta. Ohjelma oli laajennus taloushallinto-ohjelmaan, johon oli aiemmin tehty nimikekortisto, asiakasrekisteri ja osa admin-moduulia. Ohjelman toteutus jakaantui kolmeen eri osaan, suunnitteluun, toteutukseen ja testaukseen. Suunnitteluvaiheessa suunniteltiin ohjelman käyttöliittymä, tietokantarakenne ja ohjelman toiminnot. Toteutusvaiheessa toteutettiin ohjelmaa ja sen jälkeen käytiin läpi testausvaihe, jossa pyrittiin samaan suurimmat ohjelmointivirheet pois.

Lopputuloksena saatiin aikaiseksi työn alussa tavoitteena ollut laskutusmoduuli ja laajennettu admin-moduuli. Jatkossa laskutusmoduuliin tullaan tekemään loppuun ja taloushallinto-ohjelmaan tehdään ostoreskontramoduulilaajennus.

Asiasanat: C#, .NET, SQL

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Software Development

Author(s): Tuomas Reijonen
Title of thesis: Quoniam billing module
Supervisor(s): Pertti Heikkilä
Term and year when the thesis was submitted: Autumn 2011
Number of pages: 37 + 2 appendix pages

The purpose of this thesis was to create billing module to new financial management program. Billing module is SQL based desktop application which allows user to do printable or normal pdf document for client. Billing module also writes down every transaction what user does and uses those transactions for accounting. Also in this project, there were made some changes to admin-module what billing module uses.

Billing module was done by using C# as programming language and MSSQL for data storing. Billing module was extension to a financial management program what had some other modules created before, such as customer register, inventory register and admin-module. The programs execution was divided to three phases design, execution and testing. During design phase the following things were designed for the application: user interface, data storage structure and programs functions. In execution phase the program was coded and user interface created and in testing phase the program was tested to get rid of the bugs.

As a result of this project, everything what was planned at start of this project was achieved and admin-module was extended. In future the billing module will be finished and a new purchase ledger module will be made.

Keywords: C#, .NET, SQL

ALKULAUSE

Aloitin työt Nexenizerillä vuoden 2010 kesällä. Sinä kesänä aloitettiin uusi taloushallintaohjelma ja ohjelman tekoa on jatkettu siitä eteenpäin. Sain suoritettua kaksi harjoitteluprojektia ohjelmaa tehdessäni, ja samalla avautui mahdollisuus tehdä opinnäytetyö samasta ohjelmasta. Totta kai tein sitten siitä mielelläni opinnäytetyönkin kun olen työskennellyt jo ohjelman ja yrityksen kanssa niinkin pitkään.

Haluan kiittää tohtori Niko Jukaraista, että hän antoi mahdollisuuden tehdä kaikki nämä harjoittelut ja opinnäytetyön hänen yritykselleen.

Opinnäytetyön ohjaajana toimi lehtori Pertti Heikkilä.

Oulussa 23.11.2011

Tuomas Reijonen

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	6
LYHENTEET JA TERMISTÖ	8
1 JOHDANTO	9
2 PROJEKTIN LÄHTÖKOHTA	10
3 LASKUTUSMODUULI	11
3.1 Ohjelman käyttöliittymä	11
3.2 Ohjelman toiminnot	12
3.2.1 Laskun luonti ja tallennus	12
3.2.2 Laskun poisto, muokkaus ja tulostus	15
3.2.3 Laskun haku, selaus ja asiakastapahtumat	16
4 ADMIN-MODUULI	17
4.1 Yrityksen perustiedot	17
4.3 Tililuettelon ylläpito	19
4.4 Maksuehdot ja toimitustavat ja -ehdot	19
5 KÄYTETYT TEKNOLOGIAT	21
5.1 C#	21
5.2 .NET Framework	21
5.3 PDFsharp	21
5.4 Microsoft SQL Server 2008	22
6 LASKUTUSMODUULIN TOTEUTUS	23
6.1 Uuden laskun luominen ja tallentaminen	23
6.2 Laskun poisto- ja muokkaustoiminnot	24
6.3 Laskun tulostustoiminto	26
6.4 Laskun haku, selaus ja asiakastapahtumat	28
7 ADMIN-MODUULIN TOTEUTUS	31
7.1 Logon luonti	31
7.2 Yrityksen perustietojen tallennus	32
7.3 Tililuettelon ylläpito, maksuehdot, toimitustavat ja -ehdot	32
8 TESTAUS	34

<u>9 JATKOKEHITYSMAHDOLLISUUDET.....</u>	<u>35</u>
<u>10 POHDINTA.....</u>	<u>36</u>
<u>LÄHDELUETTELO.....</u>	<u>37</u>

LYHENTEET JA TERMISTÖ

C#	Microsoftin .NET-konseptia varten kehittämä ohjelmointikieli
DGV	Datagridview, kontrolli jossa voidaan näyttää ja muokata dataa
ID	Identification, tunniste
MSSQL	Microsoftin kehittämä relaatiotietokanta
PDF	Portable Document Format, Adoben kehittämä siirrettävä tiedostomuoto
SQL	IBM:n kehittämä kyselykieli, jolla voi tehdä hakuja, muutoksia ja lisäyksiä relaatiotietokantaan
.NET	Microsoftin kehittämä ohjelmistokomponenttikirjasto

1 JOHDANTO

Quoniam on täysin uusi taloushallintaohjelma, joka käyttää hyväkseen MSSQL-palvelinta tiedon säilytykseen. Idea uuteen taloushallintaohjelmaan on lähtenyt siitä, kun on huomattu puutteita nykyisissä saatavilla olevista taloushallintaohjelmista. Myös markkinoilla on tilaa ja kysyntää uudelle taloushallintaohjelmalle.

Tämän opinnäytetyön aihe on laskutusmoduuli, joka on osa suurempaa taloushallintaohjelmistokokonaisuutta, jonka nimi kirjoitushetkellä oli Quoniam. Työssä myös tehtiin lisäyksiä Quoniamin admin-moduulin, jota laskutusmoduuli käyttää hyödykseen.

Tässä dokumentissa kerrotaan, kuinka osa laskutus- ja admin-moduulista toteutettiin. Työn tavoitteena oli tehdä laskutusmoduulia mahdollisimman pitkälle versioksi, joka vastaisi tilaajan visiota.

2 PROJEKTIN LÄHTÖKOHTA

Tämän opinnäytetyön aloittaessani oli ohjelmistopakettiin tehty valmiiksi asiakasrekisteri, nimikekortisto ja osa admin-moduulista, joita laskutusmoduuli tarvitsee toimiakseen. Itse laskutusmoduuliin oli tehty peruskäyttöliittymä ja tietokanta, jota se käyttää. Toiminnallisuuksiltaan laskutusmoduuli sisälsi uuden laskun lisäyksen, mutta projektin aikana se rakennettiin uudelleen paremmaksi.

Tästä lähtökohdasta moduulia lähdettiin laajentamaan seuraavasti:

- laskun luonti, muokkaus ja poistaminen
- asiakkaan ja tuotteiden tietojen haku tietokannasta laskua varten
- asiakkaan aiempien tapahtumien selaus
- mahdollisuus luoda laskusta dokumentti tulostettavaksi
- mahdollisuus tehdä viivakoodilasku.

3 LASKUTUSMODUULI

Laskutusmoduulin avulla voidaan tehdä asiakkaille laskuja ja sitä voidaan käyttää hyödyksi myös kirjanpidossa. Tässä luvussa käydään läpi, minkälainen on ohjelman käyttöliittymä, miten ohjelma toimii ja mitä toimintoja se sisältää.

3.1 Ohjelman käyttöliittymä

Laskutusmoduulin pääkäyttöliittymä näkyy kuvassa 1. Pääkäyttöliittymä tulee näkyville, kun käyttäjä käynnistää laskutusmoduulin.

The screenshot shows the 'Laskutus' (Billing) application window. The interface includes a menu bar with 'Laadinta', 'Muokkaus', and 'Asiakas'. Below the menu bar, there are input fields for 'Numero' (359), 'Päiväys' (23.8.2011), and buttons for 'Haku' (Search). The form is divided into sections for 'Laskutusasiakas' (Billing Customer) and 'Toimitusasiakas' (Delivery Customer), both with fields for 'Nimi', 'Nimi 2', 'Osoite', 'Postinumero', and 'Toimipaikka'. There are also fields for 'Tilausnumero', 'Vitteemme', 'Vitteenne', 'Toimitustapa', 'Toimitusehto', 'Valuutta' (EUR), 'Toimitus aika', 'Maksuehto', and 'Eräpäivä' (23.8.2011). A table at the bottom shows the bill items with columns: Koodi, Nimike, Määrä, Yksikkö, Ä hinta, Alennus, and Summa. The first item is '1090T' with 'AMD Phenom II X6 1090T' and a sum of 146,34. A summary table at the bottom right shows 'Veroton' (118,98), 'Alv' (27,36), and 'Yhteensä' (146,34).

Koodi	Nimike	Määrä	Yksikkö	Ä hinta	Alennus	Summa
1090T	AMD Phenom II X6 1090T	1		146,34		146,34

Veroton	EUR	118,98
Alv	EUR	27,36
Yhteensä	EUR	146,34

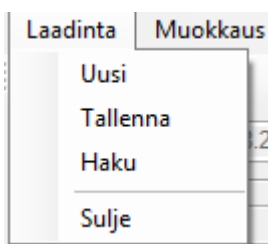
KUVA 1. Pääkäyttöliittymä

3.2 Ohjelman toiminnot

Laskutusmoduulin toiminnot jakautuvat seuraavanlaisesti: uuden laskun lisäys, olemassa olevan laskun muokkaus tai poisto, olemassa olevien laskujen selaus ja haku, asiakkaan aiempien tapahtumien selaus, tuotteiden haku nimikekortistosta, mahdollisuus valita mihin tilille tapahtuma kirjataan, toimitustavan ja -ehdon valinta ja laskun tulostus.

3.2.1 Laskun luonti ja tallennus

Laskun luonti tapahtuu painamalla työkalupalkissa olevaa Laadinta-kohtaa, josta avautuu valikko, josta voidaan luoda uusi lasku, tallentaa lasku tai hakea olemassa olevia laskuja (kuva 2).



KUVA 2. Laadinta-valikko

Kun käyttäjä painaa Uusi-kohtaa, ohjelma laskee laskulle laskun numeron, kaikki tietokentät tyhjenevät, tietokentät tulevat muokattaviksi ja tallennusmahdollisuus aktivoituu. Tietokentät, tallennus ja poistomahdollisuus ovat aina lukittuina, ellei käyttäjä luo uutta laskua tai sitten valitse laskun muokkausta. Tämä on tehty sitä varten, ettei käyttäjä vahingossa selatessa tai muussa tapauksessa tee muutoksia olemassa oleviin laskuihin.

Ohjelma myös lataa oletusarvot toimitustavalle ja -ehdolle, maksuehdolle, valuutalle, kirjaustavalle ja Debet-tilille ja hakee nykyisen päivän päiväyskenttään, kun käyttäjä luo uuden laskun.

Laskua aloitetaan yleensä tekemään ensin täyttämällä laskutus- ja toimitusasiakkaan tiedot (kuva 3). Käyttäjä voi täyttää itse tiedot tai hakea asiakkaan tiedot asiakasrekisteristä, joko asiakasnumerolla tai nimellä. Jos hakukentän jättää tyhjäksi, ohjelma hakee kaikki asiakkaat rekisteristä. Kun hakupainiketta painetaan, tulee käyttäjälle valikko, josta hän voi valita haluamansa asiakkaan (kuva 4).

Laskutusasiakas		Toimitusasiakas	
Nimi	Tuomas Reijonen	Nimi	Tuomas Reijonen
Nimi 2		Nimi 2	
Osoite	Fakestreet 1	Osoite	Fakestreet 1
Postinumero	90100	Postinumero	90100
Toimipaikka	Oulu	Toimipaikka	Oulu

KUVA 3. Laskutus- ja toimitusasiakasnäkymä

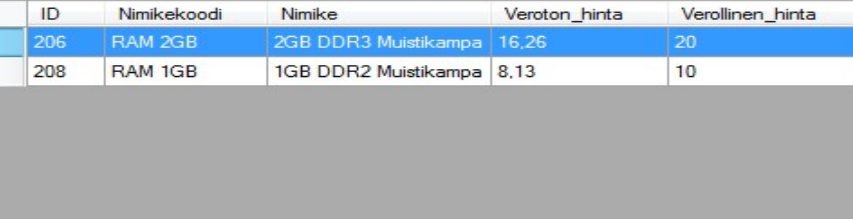
	Numero	Yritys	Lahiosoite	Postitoimipaikka	Postinumero	Puhelin
▶	112	Tuomas Reijonen	Fakestreet 1	Oulu	90100	044-654321
	113	Test Dummy	Test street	Debug	101110	050-112345

KUVA 4. Asiakashaku

Asiakastietojen täyttämisen jälkeen käyttäjä täyttää muut tiedot tarvittaessa, kuten tilausnumero, viitteet, toimitustavat ja ehdot. Käyttäjä voi myös tietyissä asiakastapahtumissa valita kirjaustavan verottomaksi ja valita mieleisensä tilityypin, johon suoritus kirjataan.

	Koodi	Nimike	Määrä	Yksikkö	Ä hinta	Alennus	Summa
	1090T	AMD Phenom II X6 1090T	1		146,34		146,34
	RAM 2GB	2GB DDR3 Muistikampa	1		20,00		20,00
**							

Veroton	EUR	135,24
Alv	EUR	31,10
Yhteensä	EUR	166,34



The screenshot shows a software window titled "Tuotehaku" (Product Search). It contains a table with the following columns: ID, Nimikekoodi (Product Code), Nimike (Product Name), Veroton_hinta (Net Price), Verollinen_hinta (Gross Price), and Valua (Value). The table lists two RAM products. The first row is highlighted in blue. Below the table is a large grey rectangular area, and at the bottom are two buttons: "Peruuta" (Cancel) and "Valitse" (Select).

	ID	Nimikekoodi	Nimike	Veroton_hinta	Verollinen_hinta	Valua
▶	206	RAM 2GB	2GB DDR3 Muistikampa	16,26	20	€
	208	RAM 1GB	1GB DDR2 Muistikampa	8,13	10	€

Peruuta Valitse

14

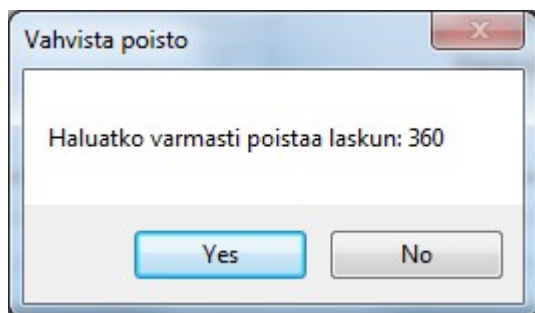
Tämän jälkeen käyttäjä voi tallentaa laskun painamalla ohjelman yläosassa olevaa tallennuspainiketta (kuva 7) tai painamalla Laadinta-valikosta (kuva 2). Jos käyttäjä haluaa peruuttaa laskun, hän voi painaa takaisin-nuolta tai poista-painiketta (kuva 7).



KUVA 7. Luo uusi lasku-, tallenna-, poista-, tulosta- ja selaus-painikkeet

3.2.2 Laskun poisto, muokkaus ja tulostus

Käyttäjä voi poistaa haluamansa laskun painamalla poista-painiketta (kuva 7) tai valita poista toiminnon Muokkaus-valikon alta. Tämän jälkeen käyttäjälle tulee vahvistuspyyntö laskun poistamisesta (kuva 8).



KUVA 8. Poistamisen vahvistuspyyntö

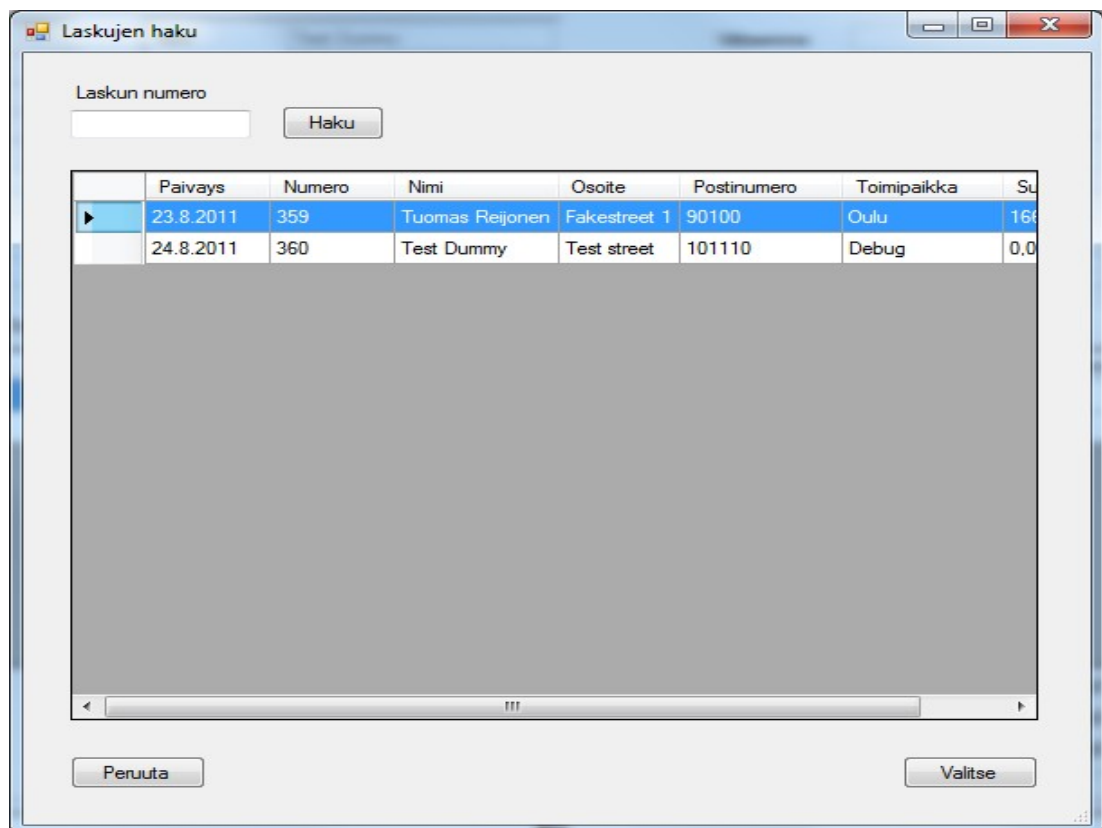
Laskunmuokkaustoiminto löytyy Muokkaus-valikon alta. Muokkaustoiminto avaa käyttäjälle muokattavaksi tai poistettavaksi laskun, joka on muuten lukittuna. Käyttäjä voi muokata mitä tahansa laskusta paitsi laskun numeroa.

Käyttäjä voi tulostaa laskun painamalla tulosta-painiketta, joka löytyy samalta riviltä, missä on tallenna-, poista- ja muuta-painikkeet. Tämän jälkeen käyttäjä voi valita, minkätyyppisen laskun hän haluaa tulostaa, esimerkiksi käteis-laskun (liite 1) tai viivakoodillisen laskun (liite 2). Tämän jälkeen

ohjelma avaa vielä esikatselunäkymän, jossa voidaan tarkistaa, että lasku on halutunlainen, ja tulostaa se.

3.2.3 Laskun haku, selaus ja asiakastapahtumat

Laskuja käyttäjä voi hakea kahdella tavalla, joko selaamalla niitä yksitellen nuolipainikkeilla (kuva 7) tai käyttämällä laskujen hakutoimintoa. Laskujen hakutoiminnolla käyttäjä voi selata nopeammin olemassa olevia laskuja tai vaihtoehtoisesti etsiä niitä laskun numerolla (kuva 9). Laskujen hakutoiminto löytyy Laadinta-valikon alta.



KUVA 9. Laskujen hakunäkymä

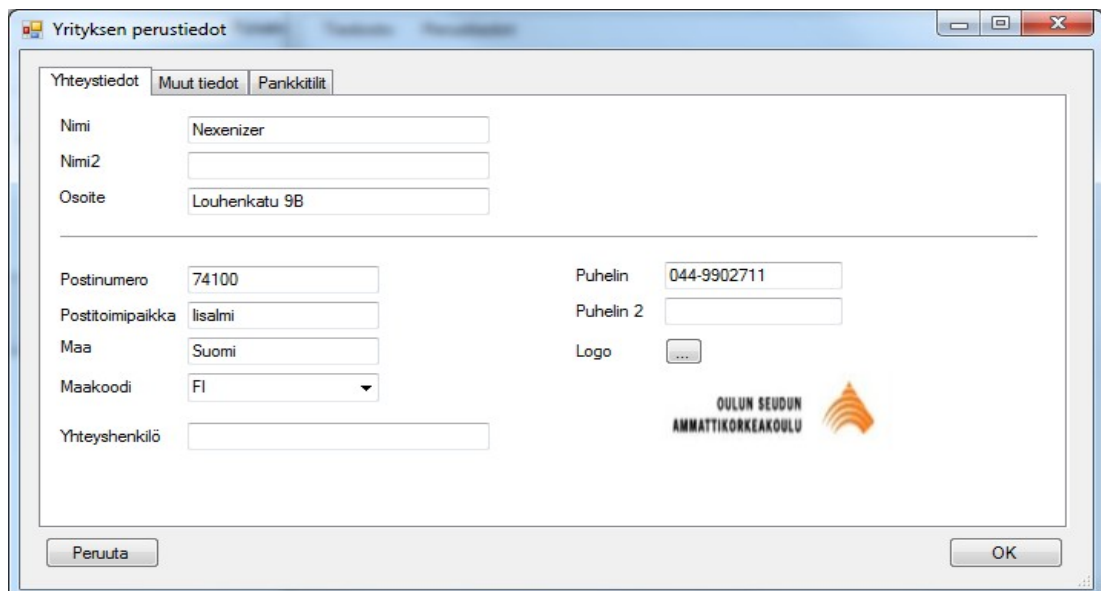
4 ADMIN-MODUULI

Admin-moduuli on koko ohjelman sovellus. Sen avulla käynnistetään muut ohjelmiston moduulit ja luodaan tietokanta ohjelmaa varten. Sen avulla myös hallinnoidaan toimitustapoja ja ehtoja, maksuehtoja, tilejä ja yrityksen perustietoja. Tässä luvussa käydään läpi, miltä tässä projektissa admin-moduulin tehdyt lisäykset näyttävät ja mitkä niiden tarkoitukset ovat.

4.1 Yrityksen perustiedot

Yrityksen perustiedoissa käyttäjä voi hallinnoida yrityksen perustietoja yhdessä ikkunassa. Ikkunaan on tehty sivujako käyttöliittymän selkeyttämiseksi ja tietojen jakamiseksi omiin osa-alueisiin.

Ensimmäisellä sivulla voi hallinnoida yrityksen perustietoja, kuten nimi, logo, puhelinnumero ja osoite (kuva 10).



KUVA 10. Yrityksen yhteystiedot ja logo

Vaihtamalla sivua voidaan täyttää yrityksen muita tietoja, kuten Alv-tunnus, Y-tunnus, sähköposti, kotisivu jne. (kuva 11).

Yrityksen perustiedot

Yhteystiedot Muut tiedot Pankkitilit

Alv-tunnus Kotipaikka

Y-tunnus OVT-tunnus

Finvoice-osoite

Finvoice-välittäjä

Sähköposti

Kotisivu

Peruuta OK

KUVA 11. Yrityksen muut tiedot

Viimeinen asia, mitä käyttäjä voi hallinnoida yrityksen perustiedoissa, on pankkitilit. Pankkitilien hallintasivu löytyy viimeiseltä sivulta, mistä löytyy käyttöliittymä tilien hallintaan (kuva 12).

Yrityksen perustiedot

Yhteystiedot Muut tiedot Pankkitilit

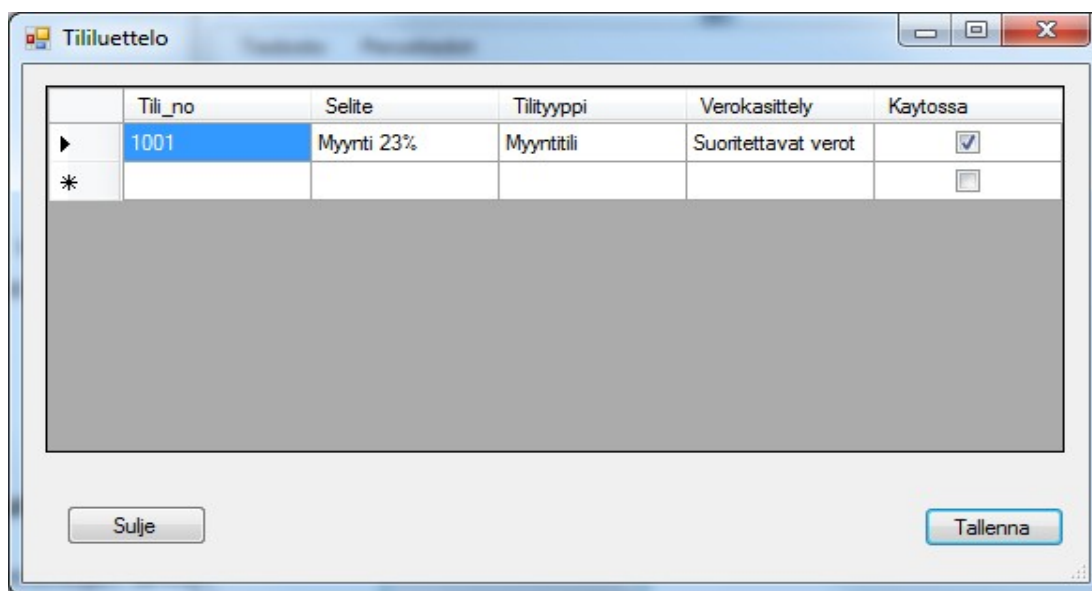
	Kotim	Tilinumero	IBAN	BIC	Saldo	Valuutta
▶	<input checked="" type="checkbox"/>	123456-13245	FI4447891	HELSFIHH		
* *	<input type="checkbox"/>					

Peruuta OK

KUVA 12. Pankkitilien hallinnointi

4.3 Tililuettelon ylläpito

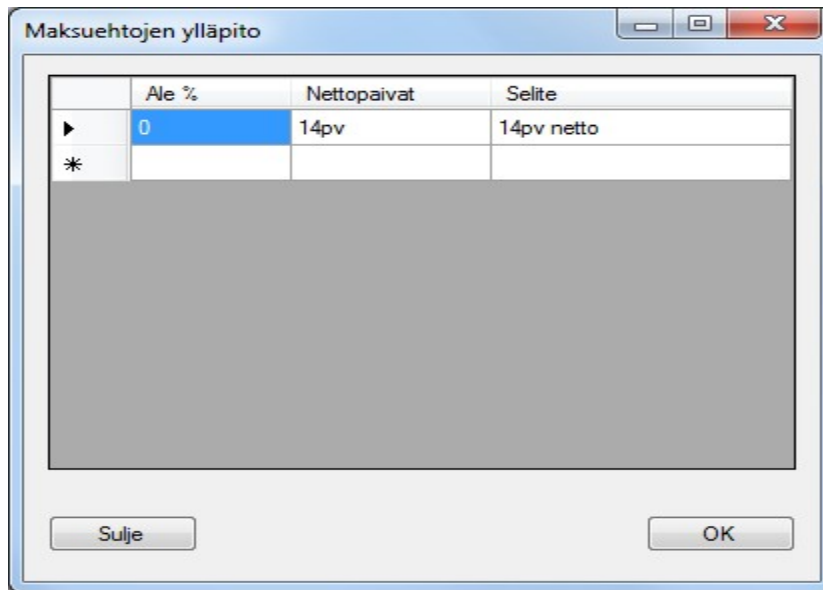
Tililuettelo on käytettävien kirjanpitotilien luettelo ja sitä käytetään kirjanpidossa. Käyttäjä voi hallinnoida sitä tililuettelon ylläpidon avulla (kuva 13).



KUVA 13. Tililuettelon ylläpidon käyttöliittymä

4.4 Maksuehdot ja toimitustavat ja -ehdot

Maksuehtojen ja toimitustapojen ja -ehtojen tarkoituksena on helpottaa laskujen tekoja. Lisättyä uutta maksuehtoa voidaan esimerkiksi hyödyntää laskun teossa siten, että käyttäjä valitsee sen maksuehto-valikosta ja siihen liittyvät tiedot täydentyvät laskuun automaattisesti. Näin käyttäjältä säästyy aikaa ja laskun tekeminen on tehokkaampaa. Kuvassa 14 näkyy maksuehtojen ylläpidon näkymä.



KUVA 14. Maksuehtojen ylläpitonäkymä

5 KÄYTETYT TEKNOLOGIAT

5.1 C#

C# on Microsoftin .NET konseptia varten kehittämä oliopohjainen ohjelmointikieli. Kieli kehitettiin yhdistämällä C++:n tehokkuus ja Java-ohjelmointikielen helppous. C#-kääntäjä kääntää C#-ohjelmat tavukoodimuotoon, ja vasta kun ohjelmaa ajetaan, käännetään ohjelman koodi koneen natiiville kielelle. Tavukoodin avulla ohjelmia voidaan siirtää eri alustoille ilman uutta kääntämistä. (1.)

5.2 .NET Framework

.NET Framework on Windowsin sisäinen komponentti, joka tukee seuraavan sukupolven ohjelmien ajoa ja kääntöä. .NET Frameworkissa on kaksi keskeistä osaa, Common Language Runtime ja .NET Framework-luokkakirjasto. Common Language Runtime on .NET Frameworkin perusta. Runtimea voidaan ajatella agenttina, joka hallinnoi koodia suorituksen aikana, hoitaa muistin hallinnan, hoitaa säikeistyksen ja samalla parantaa koodin turvallisuutta ja luotettavuutta. .NET Framework-luokkakirjasto on kokoelma uudelleenkäytettäviä luokkia, arvoja ja käyttöliittymiä, jotka ovat integroitu Common Language Runtimeen. (2.)

5.3 PDFsharp

PDFsharp on ilmainen ja vapaasti käytettävä .NET-kirjasto, jonka avulla voidaan tehdä PDF-tiedostoja (3).

5.4 Microsoft SQL Server 2008

Microsoft SQL Server on elokuussa 2008 Microsoftin julkaisema SQL-palvelin, jonka tarkoitus on säilyttää ja noutaa tietoja muiden ohjelmien pyynnöstä. Tietoja voidaan tallentaa tai noutaa joko lokaalisti tai verkon yli. Microsoftin SQL-palvelimien pääkyselykielet ovat T-SQL ja ANSI-SQL. (4.)

6 LASKUTUSMODUULIN TOTEUTUS

Tässä luvussa käydään läpi, miten laskutusmoduulin toiminnot on toteutettu, ja kuvataan niiden toimintoja tarkemmin. Laskutusmoduuli toteutettiin käyttäen C#-ohjelmointikieltä.

6.1 Uuden laskun luominen ja tallentaminen

Ensimmäisenä ohjelmaan toteutettiin uuden laskun luonti. Uuden laskun tiedot tallennetaan kahteen eri SQL-tauluun. Ensimmäinen taulu on info-taulu ja se on tarkoitettu laskun numerolle, asiakkaan tiedoille, viitteille, päivämäärälle jne. Toinen taulu on Products-taulu ja se on ostettavia tuotteita varten. Nämä kaksi taulua on linkitetty laskun numerolla toisiinsa, jotta lasku saadaan haettua uudestaan vaikka selausta varten. Laskun numeron avulla saadaan haettua oikeat tiedot ja tuotteet tapahtumalle.

Taulujen linkityksessä tuli ongelmaksi se, että taulujen primary-avaimien täytyy olla yksilöllisiä. Ongelmia tuli, jos asiakkaalla oli laskussa useampia tuotteita. Tätä ongelmaa ei ollut info-tilujen puolella, koska siellä ei tarvittu kuin yksi primary-avain laskua kohden.

Ratkaisin linkitysongelman lisäämällä kaksi kirjainta väliltä A–Z laskun numeron perään, kun käyttäjä lisää uuden tuotteen, eli esim. 100AD. Kirjaimen lisäys tapahtuu DGV:hen eventissä, kun solua ruvetaan muokkaamaan (kuva 15). Ensin ohjelma tarkistaa, onko kyseisellä rivillä vielä mitään tietoa, ja sen jälkeen lisää sille tarvittavan tunnisteen. Jos tunniste on saavuttanut kirjaimen Z, ohjelma korottaa ensimmäistä kirjainta seuraavaan esim. AZ -> BA. Tämä mahdollistaa 625 eri tuotetta asiakkaalle, mikä riittää hyvin.

```

if (Product_dataGridView.CurrentRow.Cells[0].Value == DBNull.Value)
{
    if (letter2 <= alphaEnd)
    {
        letter2++;
    }

    if (letter2 == alphaEnd)
    {
        letter2 = alphaStart;
        letter1++;
    }

    Product_dataGridView.CurrentRow.Cells[0].Value = Convert.ToString(ID_toolStripTextBox.Text) + letter1 + letter2;
}

```

KUVA 15. Tunnisteen laskeminen

Uuden laskun toteutuksessa täytyi ottaa myös huomioon kaikkien tarvittavien kenttien ja tietojen alustukset ja uuden laskun numeron laskeminen.

6.2 Laskun poisto- ja muokkaustoiminnot

Kun lasku poistetaan, käy ohjelma läpi kolme eri vaihetta. Ensimmäiseksi poistetaan poistettavat rivit tauluista. Info-aulun rivit poistetaan perinteisellä tavalla ottamalla yhteys tietokantaan ja antamalla sille poistokysely. Tuotetaulusta poistetaan rivit siten, että rivit poistetaan datasetistä ja sen jälkeen tauluadapteri päivitetään datasetillä, josta rivit on poistettu (kuva 16).

```

string deleteQuery = "DELETE FROM QNM_BIL_INFO WHERE Numero = '" + ID_toolStripTextBox.Text + "'";
SqlCommand Command = new SqlCommand(deleteQuery);
Command.Connection = Connection;

Connection.Open();
Command.ExecuteNonQuery();
Command.Connection.Close();

int row = quoniamDataSet.QNM_BIL_BILL.Rows.Count;

while (row > 0)
{
    row--;
    quoniamDataSet.Tables["QNM_BIL_BILL"].Rows[row].Delete();
}

this.qNM_BIL_BILLTableAdapter.Update(this.quoniamDataSet.QNM_BIL_BILL);

```

KUVA 16. Rivien poistot tauluista

Kun lasku on poistettu, ohjelma hakee poistetun viereisen laskun ohjelmaan näkyville. Jos laskuja ei ole enää taulussa, ohjelma tyhjentää tarvittavat kentät ja jää odottamaan uuden laskun syöttämistä.

Viimeisessä vaiheessa ohjelma laskee, mitkä ovat maksimi- ja minimilaskujen ID:t, jotta laskujen selaus toimisi oikein. Jos esimerkiksi maksimilasku olisi poistettu taulusta, osaisi ohjelma lopettaa selauksen oikeassa kohdassa.

Kun laskua halutaan muokata, ohjelma aktivoi kaikki kentät ja tarvittavat painikkeet käyttöön (kuva 17).

```
foreach (Control ctrl in this.Controls)
{
    if (ctrl.GetType() == typeof(Button))
    {
        ctrl.Enabled = true;
    }

    if (ctrl.GetType() == typeof(TextBox))
    {
        ctrl.Enabled = true;
    }

    if (ctrl.GetType() == typeof(ComboBox))
    {
        ctrl.Enabled = true;
    }
}

foreach (Control ctrl in panell1.Controls)
{
    if (ctrl.GetType() == typeof(TextBox))
    {
        ctrl.Enabled = true;
    }
}

foreach (Control ctrl in panel2.Controls)
{
    if (ctrl.GetType() == typeof(TextBox))
    {
        ctrl.Enabled = true;
    }
}
```

KUVA 17. Painikkeiden ja kenttien aktivointi muokattavaksi

6.3 Laskun tulostustoiminto

Laskun tulostus käyttää hyväkseen PDFSharp-kirjastoa, jonka avulla saadaan tehtyä PDF-dokumentteja tulostettavaksi. Kun luodaan uusi PDF-dokumentti, lisätään dokumenttiin tyhjä sivu ja luodaan tarvittavat piirto- ja kirjoitusobjektit. Samalla luodaan myös kuvaobjekti logoa varten (kuva 18).

```
//Create a new PDF document
PdfDocument document = new PdfDocument();
document.Info.Title = "Lasku" + ID_toolStripTextBox.Text;

//Create an empty page
PdfPage page = document.AddPage();

//Get an XGraphics object for drawing
XGraphics gfx = XGraphics.FromPdfPage(page);

//Objects for writing
XPen pen = new XPen(XColors.Black, 0.8);
XPen bold_pen = new XPen(XColors.Black, 1.5);
XFont small_font = new XFont("Arial", 8, XFontStyle.Regular);

XImage logo = XImage.FromFile(@"logo.jpeg");
```

KUVA 18. Uuden PDF-dokumentin luominen

Tämän jälkeen piirretään laskun yläosaan laatikot ja sijoitetaan laatikoihin oikeat tekstit kuten laskun numero, viite jne. Samalla tehdään myös laskun osat, jossa näkyy saate, tuotteet, viitenumero, kokonaishinta ja yrityksen tiedot (kuva 19).



Lasku

Test Dummy Test street 101110 Debug	Päiväys	24.8.2011	Lasku	360
	Viitteenne	555		
	Laskutusasiakas	113	Viitteemme	444
	Toimitusaika		Toimitustapa	Nouto
	Maksuehto	Käteinen	Eräpäivä	24.8.2011
	Huom. aika		Viiv. korko	13%
	Tilausnumero	123		

Saate

Koodi	Nimike	Määrä yks	ä hinta	Ale %	Alv %	Yhteensä
RAM 2GB	2GB DDR3 Muistikampa ahvenanmaan varastosta	1	10,00		23	10,00
						10,00
		Alv %	Netto	Vero	Brutto	
		23%	8,13	1,87	10,00	

		Viitenumero 360001134	Yhteensä	EUR	10,00
Nexenizer Louhenkatu 9B 74100 Test street Suomi	Puhelin 044-9902711	Pankit 123456-13245 FI4447691 HELSINKI	Y-Tunnus Kotipaikka		

KUVA 19. Laskun yläosa

Koska saatteen ja tuotteen nimikkeen tai koodien merkkijonot pystyvät olemaan pitkiä, pitää merkkijono katkaista aina oikealla kohdalla, jotta merkkijono ei mene muiden tekstien päälle tai osa tekstistä jäisi näkemättä. Tätä varten ohjelma laskee, kuinka pitkä merkkijono on, katkaisee sen tarvittaessa osiin ja sijoittaa osan tekstistä toisille riveille. Tämä vaikuttaa aina myös laskun muihin osiin, jotka ovat lisättävän rivin alapuolella, jolloin niidenkin pitää siirtyä aina tietyn verran Y-akselilla.

Jos lasku ei ole käteislasku, tehdään laskuun erillinen loppuosa, jossa on normaaliin laskuun tarvittavat asiat, kuten viivakoodi (kuva 20).

Saajan tilinumero Mottagarens kontonummer	123456-13245	IBAN	FI4447891	BIC	HELSFIHH
Saaja Mottagaren	Nexenizer Louhenkatu 9B 74100 Iisalmi				
Maksajan nimi ja osoite Betälarens namn och adress	Test Dummy Test street 101110 Debug				
Allekirjoitus Underskrift		Viitenro Ref.nr	360001134		
		Eräpäivä Förf.dag	24.8.2011	EUR	10,00



Maksu välitetään saajalle vain Suomessa kotimaan
maksuväilyksen yleisten ehtojen mukaisesti ja vain
maksajan ilmoittaman tilinumeron perusteella.
Betalingen förmedlas till mottagare endast i Finland
enligt Allmänna villkor för inrikes betalningsförmedling
och endast till det kontonummer betalaren angivit.
PANKKI BANKEN

KUVA 20. Laskun alaosa

6.4 Laskun haku, selaus ja asiakastapahtumat

Laskun selaus toteutettiin sitomalla info-taulun DataSet textboxeihin. Tämän avulla voidaan uusi tieto päivittää helposti näkyville päivittämällä vain DataSetin sisältö. Selauksessa piti ottaa huomioon, että jos lasku on viimeinen taulussa, seuraava-painike otetaan pois käytöstä ja päinvastoin. Tämä on tehty siksi, että käyttäjä tietää, milloin on saavuttanut viimeisen tai ensimmäisen laskun. Oli myös huomioitava, että ohjelma hyppäisi oikeaan laskuun myös silloin, kun välistä on poistettu laskuja eivätkä ne etene numerojärjestyksessä. Kuvassa 21 näkyy osa selauksen toteutuksesta, kuten tiedon haku taulusta ja sen sitominen DataSetiin.

```

if (max > num)
{
    ID_toolStripTextBox.Text = "";

    while (ID_toolStripTextBox.Text == "" && max > num)
    {
        num = num + 1;

        Adapter.SelectCommand.Connection.Close();

        Adapter.SelectCommand.CommandText =
            "SELECT * FROM QNM_BIL_INFO WHERE Numero LIKE '%" +
            num + "%'";

        Adapter.SelectCommand.Connection.Open();
        Adapter.SelectCommand.ExecuteNonQuery();
        aDataSet.Clear();
        Adapter.Fill(aDataSet, "QNM_BIL_INFO");
    }
}

```

KUVA 21. Osa selauksen toteutuksesta

Tuotteiden tiedot ovat eri taulussa ja sille on oma DataSet, jota se käyttää. Uuden tiedon näyttäminen tapahtuu päivittämällä datagridviewiin sidottu DataSet. Tämän jälkeen täytyy laskea suurin taulussa oleva tuotteen tunniste, että ei tulisi päällekkäisyyksiä, kun lisätään uusia tuotteita. Jos tunnisteita olisi kaksi samanlaista, eivät tiedot voisi tallentua. Viimeiseksi asiaksi, kun tuotteet on haettu, lasketaan niiden hinnat yhteen ja näytetään summa pääkäyttöliittymässä. Kuvassa 22 näkyy, miten tuotteiden haku on toteutettu.

```

//Loads data to datagridview
private void Bill_load()
{
    quoniamDataSet.QNM_BIL_BILL.Clear();

    string SQLQuery = "SELECT * FROM QNM_BIL_BILL WHERE Numero LIKE '" + num + "[a-z]%" + "' ";

    SqlDataAdapter Adapter = new SqlDataAdapter(SQLQuery, Connection);

    DataTable dTable = new DataTable();
    Adapter.Fill(dTable);

    Adapter.Fill(quoniamDataSet.QNM_BIL_BILL);

    int row = quoniamDataSet.QNM_BIL_BILL.Rows.Count;
    row = row - 1;

    if (row == 0 || row > 0)
    {
        max_id = dTable.Rows[row][0].ToString();
        |
        //calculates correct letter for row id if one already exists
        letter2 = Convert.ToChar(max_id.Substring(max_id.Length - 1));
    }

    if (row == -1)
    {
        max_id = ID_toolStripTextBox.Text;

        letter2 = Convert.ToChar("A");
    }

    Calculate_total_sum();
}

```

KUVA 22. Tuotteiden haku

Laskun haku tapahtuu hakemalla ensin kaikki olemassa olevat laskut. Laskut näytetään omassa ikkunassaan DGV:ssä. Tämän jälkeen ohjelma odottaa, että käyttäjä valitsee haluamansa laskun ja laskun numero palautetaan laskutusmoduuliin. Kun laskutusmoduuli saa laskun numeron, se tekee uudet haut laskun numeron perusteella. Sen jälkeen ohjelma tekee samanlaiset operaatiot, jotka se tekee, kun laskua selataan, ja tuo tiedot näyttille. Samalla tarkistetaan, ettei lasku ole viimeinen tai ensimmäinen, ja sen perusteella otetaan edellinen- tai seuraava-painike pois käytöstä.

Asiakastapahtumien haku toteutettiin vertaamalla valitun asiakkaan numeroa laskutauluun ja ottamalla talteen ne laskut, jossa asiakkaan numero esiintyy. Nämä tiedot sitten näytetään erillisessä ikkunassa, jossa käyttäjä voi selata ja valita halutun laskun näytettäväksi.

7 ADMIN-MODUULIN TOTEUTUS

7.1 Logon luonti

Logokuva tuodaan ohjelmaan minkä kokoisena tahansa. Siitä tehdään uusi objekti ja kuvasta tehdään oikean kokoinen. Kun kuvan koko on käsitelty, tallennetaan käsitelty kuva ohjelman omaan kansioon käytettäväksi. (Kuva 23.)

```
try
{
    OpenFileDialog open = new OpenFileDialog();
    open.Filter = "Image Files(*.jpg; *.jpeg; *.gif; *.bmp)|*.jpg; *.jpeg; *.gif; *.bmp";

    if (open.ShowDialog() == DialogResult.OK)
    {
        Bitmap original = new Bitmap(open.FileName);

        Bitmap logo = new Bitmap(original, 210, 55);

        logo.Save(@"logo.jpg", System.Drawing.Imaging.ImageFormat.Jpeg);

        pictureBox1.Image = new Bitmap(logo);
    }
}
catch (Exception)
{
    throw new ApplicationException("Virhe ladattaessa kuvaa");
}
```

KUVA 23. Logon luonti

Kuva menettää kuvassa 23 näkyvällä tavalla hieman laatua ja algoritmiä mahdollisesti parannellaan tulevaisuudessa. Tekohetkellä kuvan laatu ei ollut prioriteetti, koska kuva menettää laatuaan joka tapauksessa, kun sitä muokataan.

Lopuksi kuvan tiedoston nimi tallennetaan ohjelman config-tiedostoon eli tiedostoon, jossa data säilyy ohjelman sammuttuakin. Kun se on tallennettu config-tiedostoon, osaa ohjelma tulevaisuudessakin hakea aina oikean logon käytettäväksi. Kuvassa 24 näkyy, kuinka kuvan tiedoston nimi tallennetaan config-tiedostoon.

```

ArrayList al = new ArrayList();
string filename = @"Config.cfg";

using (StreamReader sr = new StreamReader(filename))
{
    string line;

    while ((line = sr.ReadLine()) != null)
    {
        al.Add(line);
    }
    sr.Close();
}

using (StreamWriter sw = new StreamWriter(filename))
{
    int lineCount = 1;

    foreach (String str in al)
    {
        if (lineCount == 6)
            sw.WriteLine(" " + picture);
        else
            sw.WriteLine(str);

        ++lineCount;
    }
    sw.Close();
}

```

KUVA 24. Config- tiedostoon kirjoitus

7.2 Yrityksen perustietojen tallennus

Perustietojen tallennus tietokantaan toteutettiin seuraavanlaisesti. Koska yrityksellä ei voi olla kuin yhdet perustiedot, kuten yrityksen nimi, taulussa ei tarvitse olla kuin yksi rivi, johon tiedot tallennetaan. Ensimmäiseksi tarkastetaan, onko tietokannassa olevassa taulussa jo aikaisempaa tietoa vai onko taulu tyhjä. Riippuen siitä, onko taulussa mitään tietoa, ohjelma luo uuden rivin tai vain muokkaa olemassa olevaa riviä.

7.3 Tililuettelon ylläpito, maksuehdot, toimitustavat ja -ehdot

Tililuettelon ylläpito, maksuehdot ja toimitustavat/ehdot on kaikki toteutettu samalla tavalla. Jokaisesta on tehty oma tauluadapteri, joka on sitten linkitetty kukin omaan DGV:hen, missä tiedot näytetään. Tiedon muokkaus, poisto ja tallennus ovat näin helppoa. Kun tehdään mitä tahansa muutoksia

tauluadapterin avulla, saadaan tiedot lisättyä tai päivitettyä helposti käyttämällä vain yhtä komentoa (kuva 25).

```
this.qNM_ADMIN_DELIVERY_METHODTableAdapter.Update(this.quoniamDataSet.QNM_ADMIN_DELIVERY_METHOD);
```

KUVA 25. Tiedon tallennus-, poisto- ja muokkauskomento

Tauluadapteri sisältää SQL-kyselyt valmiiksi. Kun se on linkitetty DGV:hen, osaa tauluadapteri tehdä oikean SQL-kyselyn sen perusteella, mitä datagridviewiin on tehty.

8 TESTAUS

Ohjelman testaussuunnitelma oli, että minä testaan ohjelman toiminnot alustavasti, että ne toimivat niille odotetulla tavalla. Minä myös hoidin MSSQL palvelintestauksen ja myös kaikki SQL-tietokantaa koskevat kyselyt. Loput ohjelman testauksesta hoidettiin Nexenizerillä, esimerkiksi toimintojen laajempi testaus, käyttöliittymän testaus ja ohjelman testaus muilla tietokoneilla.

Testaus toteutettiin siten, että testasin toimintoja niin kauan, ettei niissä näyttänyt olevan mitään ohjelmointivirheitä jäljellä, ja yritin luoda samalla käyttöliittymän vastaamaan työnantajan visiota. Tämän jälkeen ohjelmanversio lähetettiin Nexenizerille, missä toinen työntekijä testasi ohjelman toimintoja, arvioi käyttöliittymän ja testasi sitä eri koneilla ja ympäristöissä.

Testauksen tuloksena löydettiin yleensä monia ohjelmointivirheitä varsinkin Nexenizerin tekemissä testeissä. Toinen käyttäjä ajattelee eritavalla ja luo uusia tilanteita, joita ohjelmoija ei tule ajatelleeksi, ja näin löydettiin uusia ohjelmointivirheitä korjattavaksi. Ohjelma sai myös arvokasta testaustulosta, miten se toimii muilla koneilla ja käyttöjärjestelmillä. Sen avulla ohjelmaa saatiin kehitettyä eteenpäin.

9 JATKOKEHITYSMAHDOLLISUUDET

Jatkokehitysmahdollisuuksia ohjelmalle on paljon ja kaikkia toiminnallisuuksia ei saatu tehtyä vielä valmiiksi. Laskutusmoduuliin tehdään tulevaisuudessa suoritustarkistus, jonka avulla voidaan seurata, onko lasku suoritettu, ja kirjata se suoritetuksi. Laskutusmoduulista puuttuu vielä käyttöliittymä tietyille toiminnoille, kuten lomakeasetuksille, mutta niille on olemassa jo toimiva teknologia.

Jatkossa ohjelmistopakettiin tehdään ostoreskontramoduuli, joka tulee olemaan viimeinen moduuli ohjelmistopakettiin, ennen kuin se laitetaan myyntiin.

10 POHDINTA

Työn tavoitteena oli lähteä laajentamaan laskutusmoduulia projektin lähtökohdassa kuvatulla tavalla. Laskutusmoduuliin saatiin tehtyä kaikki projektin lähtökohdassa mainitut toiminnot valmiiksi. Laskutusmoduulin lisäksi admin-moduuliin tehtiin uusia toimintoja, esimerkiksi toimitustapojen ja -ehtojen hallinnointi, tililuettelon ylläpito, logon tallentaminen jne.

Työ oli mielenkiintoinen, haastava ja suurimmalta osin täysin uutta minulle. Työssä onnistuin hyvin siihen nähden, että työssä oli monta tuntematonta tekijää projektia aloittaessa. Alussa ei tiennyt, kuinka kauan niiden teossa tulisi menemään tai saataisiinko niitä tehtyä ollenkaan. Suurin osa projektiin varatusta ajasta menikin uusien teknologioiden opiskeluun, niiden etsimiseen ja toimintojen toteutuksen suunnitteluun.

Projektin suurimmat ongelmat olivat, kun palvelimella olevia tietoja yritettiin tallentaa, muokata tai poistaa erilaisissa tilanteissa. Mitä pidemmälle ohjelmaa rakensi, sitä enempi tauluilla oli yhteyksiä toisiinsa, mitä piti ottaa huomioon. Muita pienempiä ongelmatilanteita syntyi erilaisia ohjelmointivirheitä, joita esiintyi Visual Studiassa ja Microsoftin SQL-palvelinpuolella.

Projektia aloittaessani olisin luullut, että laskun tulostaminen olisi ollut suurin ongelma, koska siihen tarvittiin ulkopuolinen kirjasto ja en ollut ennen tehnyt mitään semmoista. Lopulta löysin hyvän ja ilmaisen kirjaston, jonka avulla sain luotua PDF-dokumentin tulostettavaksi.

Työ jäi kesken, mutta se oli jo odotettavissa, kun projektia aloitettiin. Jos kaikki otetaan huomioon, työssä edistyi hyvin ja suurin osa laskutusmoduulista saatiin tehtyä.

LÄHDELUETTELO

1. Introduction to the C# Language and the .NET Framework. Saatavissa:
<http://msdn.microsoft.com/library/z1zx9t92>. Hakupäivä 7.11.2011.
2. .NET Framework Conceptual Overview. Saatavissa:
<http://msdn.microsoft.com/library/zw4w595w.aspx>. Hakupäivä 7.11.2011.
3. What is PDFsharp and MigraDoc?. Saatavissa:
<http://pdfsharp.net/Overview.ashx>. Hakupäivä 7.11.2011.
4. Microsoft SQL Server. 2011. Saatavissa:
http://en.wikipedia.org/wiki/Microsoft_SQL_Server. Hakupäivä 7.11.2011.

NORMAALI LASKU

LIITE 1

OULUN SEUDUN
AMMATTIKORKEAKOULU



Tuomas Reijonen

Fakestreet 1
90100 Oulu

Lasku

Päiväys 23.8.2011	Lasku 359
Viitteenne 1111	
Laskutusasiakas 112	Viitteemme 2222
Toimitusaika	Toimitustapa Nouto
Maksuehto Käteinen	Eräpäivä 23.8.2011
Huom. aika	Viiv. korko 13%
Tilausnumero 1234	

Saate

Koodi	Nimike	Määrä yks	á hinta	Ale %	Alv %	Yhteensä
RAM 1GB	1GB DDR2 Muistikampa	1	8,13		23	8,13
1090T	AMD Phenom II X6 1090T	1	146,34		23	146,34
RAM 2GB	2GB DDR3 Muistikampa	1	20,00		23	20,00
						174,47
		Alv %	Netto	Vero	Brutto	
		23%	141,85	32,62	174,47	

Viitenumero
359001129

Yhteensä

EUR

174,47

Nexenizer
Louhenkatu 9B
74100 Fakestreet 1
Suomi

Puhelin
044-9902711

Pankit
123456-13245
FI4447891
HELSFIHH

Y-Tunnus 2135455-3
Kotipaikka Iisalmi
www.nexenizer.fi
nexenizer@nexenizer.fi

VIIVAKOODILLINEN LASKU

LIITE 2

OULUN SEUDUN
AMMATTIKORKEAKOULU



Tuomas Reijonen

Fakestreet 1
90100 Oulu

Lasku

Päiväys	23.8.2011	Lasku	359
Viitteenne	1111		
Laskutusasiakas	112	Viitteemme	2222
Toimitusaika		Toimitustapa	Nouto
Maksuehto	Käteinen	Eräpäivä	23.8.2011
Huom. aika		Viiv. korko	13%
Tilausnumero	1234		

Saate

Koodi	Nimike	Määrä yks	ä hinta	Ale %	Alv %	Yhteensä
RAM 1GB	1GB DDR2 Muistikampa	1	8,13		23	8,13
1090T	AMD Phenom II X6 1090T	1	146,34		23	146,34
RAM 2GB	2GB DDR3 Muistikampa	1	20,00		23	20,00
						174,47
				Alv %	Netto	Vero
				23%	141,85	32,62
						Brutto
						174,47

Viitenumero	359001129	Yhteensä	EUR	174,47
Nexenizer	Puhelin	Pankit	Y-Tunnus	2135455-3
Louhenkatu 9B	044-9902711	123456-13245	Kotipaikka	lissalmi
74100 Fakestreet 1		FI4447891	www.nexenizer.fi	
Suomi		HELSFIHH	nexenizer@nexenizer.fi	

Saajan tilinumero	123456-13245	IBAN	FI4447891	BIC	HELSFIHH
Mottagarens kontonummer					
Saaja Mottagaren	Nexenizer Louhenkatu 9B 74100 lissalmi				
Maksajan nimi ja osoite	Tuomas Reijonen Fakestreet 1 90100 Oulu				
Betalarens namn och adress					
Allekirjoitus Underskrift		Viltenro Ref.nr	359001129		
		Eräpäivä Förf.dag	23.8.2011	EUR	174,47



Maksu välitetään saajalle vain Suomessa kotimaan
maksuvälityksen yleisten ehtojen mukaisesti ja vain
maksajan ilmoittaman tilinumeron perusteella.
Betaltningen försänds till mottagare endast i Finland
enligt Allmänna villkor för inrikes betalningsförmedling
och endast till det kontonummer betalaren angivit.

PANKKI BANKEN